ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 315 (2004) 405–417

Theoretical Computer Science

www.elsevier.com/locate/tcs

The *abc* conjecture and correctly rounded reciprocal square roots

Ernie Croot[a], Ren-Cang Li[b,*], Hui June Zhu[c]

[a] *Department of Mathematics, University of California, Berkeley, CA 94720, USA*
[b] *Department of Mathematics, University of Kentucky, 715 Patterson Office Tower, Lexington, KY 40506, USA*
[c] *Department of Mathematics and Statistics, McMaster University, Hamilton, Canada, L8S 4K1*

Abstract

The reciprocal square root calculation $\alpha = 1/\sqrt{x}$ is very common in scientific computations. Having a correctly rounded implementation of it is of great importance in producing numerically predictable code among today's heterogenous computing environment. Existing results suggest that to get the correctly rounded α in a floating point number system with p significant bits, we may have to compute up to $3p+1$ leading bits of α. However, numerical evidence indicates the actual number may be as small as $2p$ plus a few more bits. This paper attempts to bridge the gap by showing that this is indeed true, assuming the *abc* conjecture which is widely purported to hold. (But our results do not tell exactly how many more bits beyond the $2p$ bits, due to the fact that the constants involved in the conjecture are ineffective.) Along the way, rough bounds which are comparable to the existing ones are also proven. The technique used here is a combination of the classical Liouville's estimation and contemporary number theory.

Keywords: Correct rounding; Reciprocal square root; The *abc* conjecture; Floating point number; Algebraic number

1. Introduction

Since computers have only finite memory, any involved real numbers have to be finitely approximated, in the form of *floating point numbers* (FPNs). By default in this paper, all FPNs, unless otherwise explicitly stated, are binary and of the same type,

* Corresponding author.
E-mail addresses: ecroot@math.berkeley.edu (E. Croot), rcli@ms.uky.edu (R.-C. Li), zhu@cal.berkeley.edu (H.J. Zhu).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2004.01.013

i.e., p bits in the significand, hidden bits (if any) included. Thus an FPN x takes the form

$$x = \pm 2^{m_x} \cdot (1.x_1 \cdots x_{p-1}), \tag{1.1}$$

where m_x is the *exponent*, p is the number of *significant digits*, and $x_i \in \{0,1\}$. In most commonly used FPN systems (see [1,2]), $p = 24$ ("single"), 53 ("double"), 64 ("double-extended"), or 113 ("quad"). Since one binary digit takes one bit to store, binary digits and bits are often used indistinguishably.

Even though there are restraints upon m_x in actual FPN systems, beyond which underflow or overflow occurs, for the purpose of correct roundedness these exceptional cases may be resolved by interpreting our results with a different p other than the default one. Consequently, we impose no restraints upon m_x in this paper.

Binary FPN systems are mostly common on today's computers. But this paper can be modified in a straightforward way for FPN systems in radices other than 2, e.g., the decimal FPN as proposed in [4].

Clearly no irrational real number α can be written in the form (1.1) without rounding. Without loss of generality, from now on we assume $\alpha > 0$ (which evidently holds for the reciprocal square root to be discussed soon) and write

$$\alpha = 2^{m_\alpha} \cdot (1.y_1 \cdots y_{p-1} y_p \cdots), \tag{1.2}$$

where $y_i \in \{0,1\}$. The IEEE standard mandates four *rounding modes* (see [1]). They are: *rounding to nearest* (or to nearest even whenever in a tie), *rounding towards* $+\infty$, *rounding towards* $-\infty$, and *rounding towards* 0. Under the rounding to nearest mode, correctly rounded α is

$$2^{m_\alpha} \cdot (1.y_1 \cdots y_{p-1}) + \begin{cases} 0 & \text{if } \zeta < 1/2, \text{ or } \zeta = 1/2,\ y_{p-1} = 0, \\ 2^{-p+1+m_\alpha} & \text{if } \zeta > 1/2, \text{ or } \zeta = 1/2,\ y_{p-1} = 1, \end{cases}$$

where $\zeta := 0.y_p \cdots$. Notice $\zeta = \frac{1}{2}$ does not occur as α is irrational, but we include the case merely for completeness. Under the last three modes (collectively termed the *direct rounding modes*), correctly rounded α is

$$2^{m_\alpha} \cdot (1.y_1 \cdots y_{p-1}) \text{ or } 2^{m_\alpha} \cdot (1.y_1 \cdots y_{p-1}) + 2^{-p+1+m_\alpha}$$

depending on whether $\zeta > 0$ or not. Therefore, finding the correctly rounded α rests on correct estimation of ζ. Our central concern is the following question, which is of practical and theoretical importance in computer arithmetic. See existing results in [8,11].

Question 1.1. *Given p and an irrational algebraic number α by its minimal polynomial, find the minimal number of correct leading significant bits of α in (1.2) that is necessary to round α correctly to an FPN of p significant digits.*

Let

$$n := \begin{cases} p+1 & \text{in the nearest rounding mode,} \\ p & \text{in the direct rounding modes.} \end{cases} \tag{1.3}$$

When the binary representation (1.2) of α contains k consecutive 0's (resp., 1's) starting at the bit of y_n we say it has a 0-*chain* (resp., 1-*chain*) of length k. These can be seen that the worst scenarios for rounding correctly only possibly happen when there is a long 0-chain or 1-chain. Specifically, they are in the nearest rounding mode $y_p = 1$ (resp., $y_p = 0$) followed by a long 0-chain (resp., 1-chain), and in the direct rounding mode any y_{p-1} followed by a long 0-chain or 1-chain. In both cases we may write

$$\alpha = 2^{m_\alpha} \cdot (Y_n + 2^{-n+1}\varepsilon), \quad Y_n = 1.y_1 \cdots y_{n-1}, \tag{1.4}$$

where $|\varepsilon| < \frac{1}{2}$ and $y_p = 1$ if in the nearest rounding mode.

Some of the y_i's in (1.4) may differ from those in (1.2). However, this will not affect our argument in the rest of this paper as long as their m_α's are equal, which is indeed the case with possible exceptions

$$Y_n = 2 - 2^{-n+1} \quad \text{and} \quad 0 \leqslant \varepsilon < 1/2. \tag{1.5}$$

But these exceptions can be handled individually. For the reciprocal square root case, no such cases are possible because then $x = 1/\alpha^2$ would not be any FPN. So we shall assume for the remainder of the paper that the m_α in (1.4) and (1.2) are the same.

The length of the 0-chain (or 1-chain) in α, denoted by $D(\alpha)$, is equal to

$$D(\alpha) := -\lfloor \log_2 |\varepsilon| \rfloor - 1, \tag{1.6}$$

where $\lfloor \cdot \rfloor$ means the floor of a real number. Note that $\varepsilon > 0$ (resp., < 0) corresponds to the case with a long 0-chain (resp., 1-chain). Then we find that

$$q := p + 1 + D(\alpha) \tag{1.7}$$

leading correct significant digits are sufficient for resolving Question 1.1 and hence it suffices to give an upper bound on $D(\alpha)$. This will be the subject of the paper.

Our main results concern Question 1.1 for reciprocal square root as an illustrative example. The technical details can be modified for other algebraic numbers like the cube root or powers of other fractions and their reciprocals. But we shall omit the detail because of the similarity in technicality. There is another reason for our choice of the reciprocal square root, too. It is ubiquitous in scientific computations, and has become part of the elementary function libraries `libm` provided by major computer vendors such as HP [12], IBM [9], and Intel.[1] Both HP and IBM name it `rsqrt`, while Intel names it `invsqrt`. Owing to the speed considerations, HP's and Intel's reciprocal square root subroutines for Itanium are not guaranteed to be correctly rounded, except for IEEE single precision ones. The authors are unsure about IBM's `rsqrt`, but doubted it was

[1] See `http://www.intel.com/software/products/opensource/whats_new.htm#opt_math`.

correctly rounded. In any event, it is fair to say that any of these library implementations may run roughly twice as fast as by taking[2] a square root and then a division.

The remainder of this paper is organized as follows: Our major contribution, sharper bounds on $D(\alpha)$ based on the famous *abc* conjecture (yet widely anticipated to hold) from Number Theory, are presented in Section 2, where we also proved rough bounds on $D(\alpha)$ which are comparable to the existing ones (see [8,11]). The sharper bounds can differ from the rough ones by as many as p bits. Section 3 presents a brief discussion of the *abc* conjecture, along with other theorems, that relate to the approximation of an algebraic number by rational ones. We give our conclusion remarks in Section 4.

2. Reciprocal square root

Fix an FPN $x = 2^{m_x}(1.x_1x_2\cdots x_{p-1})$ in the standard form such that x is not an even power of 2 (otherwise $1/\sqrt{x}$ will be a power of 2 and thus an exact FPN) and $m_x = 0$ or 1. Our later theorems are stated in more general terms, i.e., without restraining m_x to either 0 or 1. This is done by making

$$\tilde{m}_x := m_x \bmod 2 \in \{0, 1\} \tag{2.1}$$

appear in bounds instead.

Since $1 < 2^{m_x}(1.x_1x_2\cdots x_{p-1}) < 4$ we have $1 < \sqrt{2^{m_x}(1.x_1x_2\cdots x_{p-1})} < 2$. Let $\alpha := 1/\sqrt{x}$. Since $\frac{1}{2} < \alpha < 1$, we have $m_\alpha = -1$ in (1.2). This α is the positive solution of the equation $f(\alpha) := 1 - x\alpha^2 = 0$. For any approximation Z to α, we have $f(Z) = f(Z) - f(\alpha) = f'(\xi)(Z - \alpha)$ and hence

$$Z - \alpha = \frac{f(Z)}{-2x\xi} \tag{2.2}$$

for some ξ between α and Z. This approach is in a similar spirit to that of Liouville's estimation for arbitrary algebraic numbers of higher orders (see [13,15]).

Recall (1.3). Set $Z = 2^{-1}Y_n$. Then $\alpha = Z + 2^{-n}\varepsilon$. We may write

$$\xi = Z + t2^{-n}\varepsilon = \alpha + (t-1)2^{-n}\varepsilon = \alpha(1+\eta)$$

for some $0 < t < 1$, and $\eta = 2^{-1}(t-1)2^{-n+1}\varepsilon/\alpha$. By (2.2) we have

$$-2^{-n}\varepsilon = \frac{1 - xZ^2}{-2x\xi},$$

$$\varepsilon = 2^{n-1}(1 - xZ^2)\frac{1}{x\xi} = 2^{n-1}(1 - xZ^2)\frac{\alpha^2}{\xi} = 2^{n-1}(1 - xZ^2)\frac{\alpha}{1+\eta},$$

$$\log_2|\varepsilon| = n - 1 + \log_2|1 - xZ^2| + \log_2\left(\frac{\alpha}{1+\eta}\right).$$

[2] A correctly rounded square root followed by a correctly rounded division does not guarantee a correctly rounded reciprocal square root.

Notice

$$\begin{aligned}\frac{\alpha}{1+\eta} &= \alpha(1-\eta+\eta^2-\cdots)\\ &= \alpha - 2^{-1}(t-1)2^{-n+1}\varepsilon+\cdots\\ &= 2^{-1}(Y_n+(2-t)2^{-n+1}\varepsilon)+\cdots\ .\end{aligned}$$

It can be seen that $Y_n \neq 1$, otherwise $x = 1/\alpha^2 = 4/(1+2^{-n+1}\varepsilon)^2$ cannot be an FPN unless $\varepsilon = 0$ which corresponds to the excluded case when x is a power of 2. Since $|\varepsilon| < \frac{1}{2}$ and $Y_n > 1$, and the exceptional case (1.5) is excluded, we have $\lfloor \log_2(\alpha/(1+\eta))\rfloor = \lfloor \log_2 \alpha\rfloor = -1$. Therefore

$$\log_2|\varepsilon| \geqslant n-2+\log_2|1-xZ^2|. \tag{2.3}$$

To bound $D(\alpha)$ it suffices to get $\min|1-xZ^2|$ (or its lower bound) for all FPNs x. In what follows, we shall bound $\min|1-xZ^2|$ in two different ways: a crude one that leads to rough bounds and more refined one that first reformulates it into an integral minimization problem and then employs the *abc* conjecture. The latter leads to sharper bounds.

2.1. Rough bounds

Theorem 2.1. *Let x be an FPN of p significant digits, $\alpha = 1/\sqrt{x}$, and let $\tilde{m}_x$ be defined as in (2.1). Then either α is an FPN, or*

$$D(\alpha) \leqslant \begin{cases} 2p+1-\tilde{m}_x & \text{in nearest rounding mode,}\\ 2p-\tilde{m}_x & \text{in direct rounding modes.}\end{cases} \tag{2.4}$$

Proof. As we remarked at the beginning of Section 2, we may assume $m_x = \tilde{m}_x$ for the purpose of this proof. Because

$$xZ^2 = 2^{m_x-2}(1.x_1\cdots x_{p-1})(1.y_1\cdots y_{n-1})^2, \tag{2.5}$$

in its fixed point binary representation, the least significant bit of xZ^2 is given by

$$2^{m_x-2}\times x_{p-1}2^{-p+1}\times(y_{n-1}2^{-n+1})^2 = x_{p-1}y_{n-1}^2\times 2^{-p-2n+1+m_x}.$$

Therefore $|1-xZ^2| \geqslant 2^{-(p+2n-1-m_x)}$, and thus

$$\log_2|\varepsilon| \geqslant n-2-(p+2n-1-m_x) = -p-n-1+m_x.$$

Eq. (2.4) is a consequence of (1.3), (1.6), and (2.3). □

Applying Theorem 2.1 to the case $n = p$, we arrive at a bound $2p-m_x$ that is better than $2n+1 = 2p+1$ in [11] but one bit worse than $2n-1 = 2p-1$ in [8] for the case $m_x = 0$.

2.2. Sharper bounds assuming the abc conjecture

We shall first reduce the computation of min $|1-xZ^2|$ to an integral approximation problem which allows us to employ the *abc* conjecture. The reader is referred to Section 3 for a discussion of it. Set

$$a=(1x_1\cdots x_{p-1})_{\text{binary}},\quad b=(1y_1\cdots y_{p-1})_{\text{binary}}.$$

It can be seen that

$$2^{p-1}<a<2^p-1. \tag{2.6}$$

Since $2^{-m_x-1}<\alpha^2=1/x<2^{-m_x}$, we have $2^{(1-m_x)/2}<2\alpha=1.y_1\cdots y_{p-1}\cdots<2^{(2-m_x)/2}$. Thus $b=2^{p-1}(1.y_1y_2\cdots y_{p-1})\geqslant 2^{p-1}(1.y_1y_2\cdots y_{p-1}\cdots)-1>2^{p-1+(1-m_x)/2}-1$. Combine the above inequalities to get

$$2^{p-(1+m_x)/2}-1<b<2^{p-m_x/2}. \tag{2.7}$$

For the nearest rounding, that is $n=p+1$ and $y_p=1$, we have

$$\begin{aligned}1-xZ^2&=1-2^{m_x-p+1}a(2^{-1-p+1}(b+1/2))^2\\&=1-2^{m_x-p+1}a(2^{-1-p}(2b+1))^2\\&=1-2^{-3p-1+m_x}a(2b+1)^2\\&=2^{-3p-1+m_x}(2^{3p+1-m_x}-a(2b+1)^2).\end{aligned} \tag{2.8}$$

For the direct rounding, that is $n=p$, we have

$$\begin{aligned}1-xZ^2&=1-2^{m_x-p+1}a(2^{-1-p+1}\,b)^2\\&=1-2^{-3p+1+m_x}ab^2\\&=2^{-3p+1+m_x}(2^{3p-1-m_x}-ab^2).\end{aligned} \tag{2.9}$$

The above discussion, (1.6), and (2.3) lead to

Lemma 2.2. *Let x be an FPN with exponent m_x, $\alpha=1/\sqrt{x}$, and let $\tilde{m}_x$ be defined as in (2.1). Then either α is an FPN or*

$$D(\alpha)\leqslant\begin{cases}2p+1-\tilde{m}_x & \\ \quad-\lfloor\log_2|2^{3p+1-\tilde{m}_x}-a(2b+1)^2|\rfloor & \text{in nearest rounding mode},\\ 2p-\tilde{m}_x & \\ \quad-\lfloor\log_2|2^{3p-1-\tilde{m}_x}-ab^2|\rfloor & \text{in direct rounding mode}.\end{cases}$$

Better bounds now rest on the solutions to the following integral minimization problems: Given p and $m_x \in \{0,1\}$, find

$$\min |2^{3p+1-m_x} - a(2b+1)^2|, \tag{2.10}$$

$$\min |2^{3p-1-m_x} - ab^2|, \tag{2.11}$$

subject to (2.6) and (2.7). Solving them will provide us with much sharper bounds on the number $D(\alpha)$ of consecutive 0's (and 1's) by Lemma 2.2. This is the place where we need help from the *abc* conjecture. The reader who is not familiar with the conjecture is referred to Section 3 before proceeding from here.

Lemma 2.3. *Assume the abc conjecture holds. Let p be a positive integer and $m_x = 0$ or 1, and let a and b be integers satisfying (2.6) and (2.7). For any $0 < \tau < 1$ there exists a positive constant C_τ such that*

$$\min |2^{3p+1-m_x} - a(2b+1)^2| \geqslant C_\tau (2^p)^{1-\tau}, \tag{2.12}$$

$$\min |2^{3p-1-m_x} - ab^2| \geqslant C_\tau (2^p)^{1-\tau}. \tag{2.13}$$

Proof. (1) We shall first prove (2.12). Let

$$d := 2^{3p+1-m_x} - a(2b+1)^2. \tag{2.14}$$

Write $a = 2^s a'$ for some odd integer a' and $0 \leqslant s < p$. Then (2.14) reduces to $d' = 2^{3p+1-m_x-s} - a'(2b+1)^2$. It can be seen that $\gcd(2^{3p+1-m_x-s}, a'(2b+1)^2, d') = 1$ since $a'(2b+1)^2$ and d' are odd integers. Write $v := \tau/4$. By the *abc* conjecture (see Conjecture 3.1), there exists a constant $A_v > 0$ such that

$$2^{3p+1-m_x-s} \leqslant A_v (\mathrm{rad}(2^{3p+1-m_x-s} a'(2b+1)^2 d'))^{1+v}. \tag{2.15}$$

It can be seen that $\mathrm{rad}(2^{3p+1-m_x-s} a'(2b+1)^2 d') \leqslant 2a'(2b+1)|d'|$. Thus we get

$$\begin{aligned} 2^{3p+1-m_x-s} &\leqslant A_v (2a'(2b+1)|d'|)^{1+v} \\ &< A_v (2 \cdot 2^{p-s} \cdot 2^{p+1}|d'|)^{1+v} \\ &= A_v (2^{2p+1-s}|d'|)^{1+v}. \end{aligned}$$

Write $B_v := A_v^{-1/(1+v)}$, the above inequality is equivalent to

$$|d'| > B_v 2^{(3p+1-m_x-s)/(1+v) - (2p+1-s)}. \tag{2.16}$$

Since $v < 1$ and $0 \leqslant s < p$, we have

$$\begin{aligned} &\frac{3p+1-m_x-s}{1+v} - (2p+1-2s) \\ &\quad > (3p+1-m_x-s)(1-v) - (2p+1-2s) \end{aligned}$$

$$= p - m_x + s - v(3p + 1 - m_x - s)$$

$$> p - 4vp - 1$$

$$= p(1 - \tau) - 1.$$

Write $C_\tau := B_v/2$, then (2.16) implies that

$$|d| = 2^s|d'| > B_v 2^{(3p+1-m_x-s)/(1+v)-(2p+1-2s)} > (B_v/2)(2^p)^{1-\tau} = C_\tau(2^p)^{(1-\tau)}.$$

By (2.14) this proves our first claim.

(2) Now we prove (2.13). Since it is similar to part (1) we shall outline our proof but omit details. Let $d := 2^{3p-1-m_x} - ab^2$. Write $d = 2^s d'$ for some odd integer d' and $0 \leqslant s < p$. Then $d' = 2^{3p-1-m_x-s} - 2^{-s}ab^2$. It is easy to see that $\gcd(2^{3p-1-m_x-s}, 2^{-s}ab^2, d') = 1$. By the *abc* conjecture, we have

$$2^{3p-1-m_x-s} \leqslant A_v(2 \cdot 2^{-s}ab|d'|)^{1+v}. \tag{2.17}$$

Write $B_v := A_v^{-1/(1+v)}$. Note that $ab < 2^{2p-m_x/2}$. By (2.17) we have

$$|d'| > B_v 2^{(3p-1-m_x-s)/(1+v)+s-1-2p+m_x/2}.$$

But

$$\frac{3p - 1 - m_x - s}{1 + v} + s - 1 - 2p + \frac{m_x}{2}$$

$$> (3p - 1 - m_x - s)(1 - v) - \left(2p + 1 - \frac{m_x}{2} - s\right)$$

$$> p - v(3p - 1 - m_x - s) - 2 - \frac{m_x}{2}$$

$$> p(1 - \tau) - \frac{5}{2}.$$

Let $C_\tau := \left(\sqrt{2}/8\right) B_v$, then we have $|d'| > B_v 2^{p(1-\tau)-5/2} = \left(\sqrt{2}/8\right) B_v(2^p)^{1-\tau} = C_\tau(2^p)^{1-\tau}$. This finishes our proof. □

Theorem 2.4. *Assume the abc conjecture holds. Let x be an FPN with exponent m_x, $\alpha = 1/\sqrt{x}$, and let $\tilde{m}_x$ be defined as in (2.1). Then either α is an FPN or for any $0 < \tau < 1$ there exists a positive constant C_τ (only depends on τ) such that*

$$D(\alpha) \leqslant \begin{cases} p + 1 - \tilde{m}_x - \lfloor \log_2 C_\tau - p\tau \rfloor & \text{in nearest rounding mode,} \\ p - \tilde{m}_x - \lfloor \log_2 C_\tau - p\tau \rfloor & \text{in direct rounding modes.} \end{cases}$$

Proof. As a consequence of Lemmas 2.2 and 2.3, we have in the rounding to nearest mode

$$D(\alpha) \leqslant 2p + 1 - \tilde{m}_x - \lfloor \log_2 C_\tau + p(1 - \tau) \rfloor = p + 1 - \tilde{m}_x - \lfloor \log_2 C_\tau - p\tau \rfloor$$

and in the direct rounding modes

$$D(\alpha) \leqslant 2p - \tilde{m}_x - \lfloor \log_2 C_\tau + p(1-\tau) \rfloor = p - \tilde{m}_x - \lfloor \log_2 C_\tau - p\tau \rfloor.$$

The proof is completed. □

Bounds in Theorem 2.4 are sharper than those in Theorem 2.1 by roughly $p(1-\tau)+\log_2 C_\tau$. From the proof, one has $C_\tau = \frac{1}{2}\, c_{\tau/4}^{-4/(\tau+4)}$ for any $\tau > 0$, where c_τ is the constant given in the *abc* conjecture. One notices that when $\tau > 0$ approaches 0, C_τ approaches $1/2c_{\tau/4}$. As c_τ is ineffective, our sharper bounds in Theorem 2.4 can only serve as a theoretical background for a better understanding of the correct roundedness of the reciprocal square root.

3. The *abc* conjecture, Roth theorem and Liouville's estimates

We shall first give a brief discussion of the *abc* conjecture, which we applied in proving Theorem 2.4. The *abc* conjecture was proposed by Masser and Oesterlé independently. There are several versions of the conjecture right now, but only the traditional one is used here. For a readable survey see [6,16] and also [7, Part D] or [10, IV, Section 7].

For any non-zero integer N, let $\mathrm{rad}(N)$ denote the *radical* of N, that is, $\mathrm{rad}(N) := \prod_{\ell | N} \ell$ where the product ranges over all distinct prime divisors of N. For instance, $\mathrm{rad}(-6) = 6$ and $\mathrm{rad}(8) = 2$.

Conjecture 3.1 (The abc Conjecture). *Given $\tau > 0$ there exists a positive number c_τ such that*

$$\max(|A|, |B|, |C|) \leqslant c_\tau \cdot \mathrm{rad}(ABC)^{1+\tau}$$

for any non-zero integers A, B, C with $\gcd(A, B, C) = 1$ and $A + B = C$.

As of today the *abc* conjecture remains one of the most famous open questions in Number Theory. If the integers A, B, C are replaced by polynomials in one variable over a field, an analogous statement of the *abc* conjecture is known to hold, thanks to Mason (see [10, p. 194]). There is a long and ever-growing list of significant consequences. Remarkably, a stronger version of the *abc* conjecture implies the famous Roth theorem [17] and the Fermat's last theorem (recently resolved by Wiles, see [18]). For these reasons, the *abc* conjecture is widely anticipated to hold among number theorists. A good starter for all these may be the survey [6].

In our application (see Lemma 2.3 and Theorem 2.4) of Conjecture 3.1, the integers A, B, and C are taken more specific forms, i.e., one of them is a power of 2, and another one is either ab^2 or $a(2b+1)^2$ with some constraints on integers a and b. It is a natural question to ask whether those structures upon A, B, and C present a constrained *abc* conjecture? It is conceivable that one should arrive at a smaller constant c_τ for a given τ in these cases. But we do not know and did not pursue further in this direction.

We shall emphasize that the conjecture does not give an effective bound, namely, one does not know how small c_τ can really be.

Below we give some number theoretical perspective of our central question. Note that Question 1.1 can be reformulated as follows:

Question 3.2. *Given p and an algebraic number α by its minimal polynomial, find the minimal q such that there is an FPN y with q correct significant digits and the rounded y is equal to the rounded α in p significant bits.*

For simplicity, we confine ourselves with the mode of rounding to nearest in this section.

Proposition 3.3. *Suppose y is rounded to an FPN of p significant digits, written as $M/2^t$, where M is an integer satisfying $2^{p-1} \leqslant M \leqslant 2^p - 1$ and $t = p - 1 - m_y$. If y is a rational approximation to α precisely enough so that*

$$\left| y - \frac{M}{2^t} \right| + |\alpha - y| < \frac{1}{2^{t+1}}. \tag{3.1}$$

Then the number of correct significant bits of y is no fewer than q as required by Question 3.2. *An exception to this is when the first $p+1$ leading significant bits of α are all ones, for which $1/2^{t+1}$ must be replaced by $1/2^{t+2}$.*

Proof. Eq. (3.1) holds for y sufficiently close to α, since $|y - M/2^t| \leqslant 1/2^{t+1}$, where the inequality is strict for y sufficiently close to α (since α is irrational). Then, by the triangle inequality, we will have $|\alpha - M/2^t| < 1/2^{t+1}$, which holds if and only if α is also rounded to $M/2^t$. We leave the exception case to the reader. □

In practice, a computer can perform such approximation task by computing more and more significant bits of α and yielding y closer and closer to α, until the inequality holds. But unless α is an algebraic number such as the reciprocal square root, in general there is no way of gauging, a priori, how precise an approximation we will need; that is, we have no way of estimating the "run time" of such a procedure for an arbitrary irrational α. In what follows we shall see what we can get from Roth's theorem (see [5, p. 30] or [17]): *Let $\delta > 0$ be an arbitrarily small real number. If u, v are integers, $v \geqslant 1$, then*

$$\left| \alpha - \frac{u}{v} \right| > \frac{c_{\delta,\alpha}}{v^{2+\delta}},$$

where $c_{\delta,\alpha} > 0$ is some (ineffective) constant and depends only on δ and α. From this we have the following:

Proposition 3.4. *Let notation be as in* Proposition 3.3. *If*

$$|y - \alpha| < \frac{c_{\delta,\alpha}}{2^{(t+1)(2+\delta)}},$$

then the number of significant digits of y is no fewer than q as required by Question 3.2.

Proof. If α is not rounded to $M/2^t$ so that (3.1) fails to hold, we will have

$$\frac{1}{2^{t+1}} \leqslant \left|\alpha - \frac{M}{2^t}\right| \leqslant \left|\frac{M}{2^t} - y\right| + |y-\alpha| < \frac{1}{2^{t+1}} + \frac{c_{\delta,\alpha}}{2^{(t+1)(2+\delta)}}.$$

Thus, for $b = \pm 1$, we will have

$$\left|\alpha - \frac{2M+b}{2^{t+1}}\right| \leqslant \frac{c_{\delta,\alpha}}{2^{(t+1)(2+\delta)}},$$

which is impossible, by Roth's Theorem. Thus, for such y, α is rounded to $M/2^t$. □

It can be seen that any FPN y with $\lceil (t+1)(2+\delta) - \log_2 c_{\delta,\alpha}\rceil + m_\alpha$ correct significant bits satisfies the condition of Proposition 3.4. By proper scaling by some power of 2, we may assume $m_\alpha = 0$. Then this implies that

$$q \leqslant \lceil (t+1)(2+\delta) - \log_2 c_{\delta,\alpha}\rceil \leqslant \lceil p(2+\delta) - \log_2 c_{\delta,\alpha}\rceil, \tag{3.2}$$

where we have used $t = p - 1 - m_y \leqslant p$. This bound is comparable to Theorem 2.4, except the constant term here $\log_2 c_{\delta,\alpha}$ which depends on α itself and hence is of little practical value.

Along the same lines we may be able to develop a similar proposition based on Liouville's theorem [3] as in Section 2.1. We leave this as an exercise to the interested readers.

4. Conclusions

We have studied the minimal number q of leading correct significant bits of the reciprocal square root $\alpha = 1/\sqrt{x}$ over entire range of an FPN system enough for correctly rounding α according to the IEEE standards. The technique used is a combination of the ancient Liouville's estimation and the modern number theory. The main results are summarized in Theorems 2.1 and 2.4 which provides much sharper estimates. However, the sharper bounds are only of theoretical interest for now as they are built upon an unproven, though widely thought to be true, famous *abc* conjecture. Even so the effort here represents a step forward in bridging the gap between the existing results on the minimal number q and the numerically observed one.

Our study here on the reciprocal square root, to certain extent, is representative to other algebraic functions in scientific computations, most notably the cube root $x^{\frac{1}{3}}$ which has been included in `libm` by HP [6], Intel (see the web page in a previous footnote), and IBM [7]. It can be proved that: *Assume the* abc *conjecture holds. Let x be an FPN, $\alpha = x^{\frac{1}{3}}$. Then either α is an FPN or for any $0 < \tau < 1$ there exists a positive constant C_τ (only depends on τ) such that $D(\alpha) \leqslant p[1+\tau] + C_\tau$.* In view of similar technicality and keeping this paper short, we omit the detail.

Our focus on the binary FPN systems is representative, too. Extensions to FPN systems in radix other than 2 can be done along similar lines to what we have here.

Acknowledgements

The authors are grateful to the anonymous referees for detailed and extremely helpful comments and suggestions.

The work of Ren-Cang Li was supported in part by the National Science Foundation under Grant No. ACI-9721388 and by the National Science Foundation CAREER award under Grant No. CCR-9875201. Part of this work was conceived while he was on leave at Hewlett-Packard Company. He is grateful for help received from Jim Thomas, Jon Okada, and Peter Markstein of HP Itanium floating point and elementary math library team at Cupertino, California.

The research of Hui June Zhu was done while a postdoctoral fellow at University of California at Berkeley, and she was partially supported by a grant from the David and Lucile Packard Foundation to Bjorn Poonen of University of California at Berkeley.

References

[1] American National Standards Institute and Institute of Electrical and Electronic Engineers, IEEE standard for binary floating-point arithmetic, ANSI/IEEE Standard, Std 754-1985, New York.
[2] American National Standards Institute and Institute of Electrical and Electronic Engineers, IEEE standard for radix independent floating-point arithmetic, ANSI/IEEE Standard, Std 854-1987, New York.
[3] A. Baker, Transcendental Number Theory, 2nd Edition, Cambridge University Press, Cambridge, 1979.
[4] M.F. Cowlishaw, E.M. Schwarz, R.M. Smith, C.F. Webb, A decimal floating-point specification, in: N. Burgess, L. Ciminiera (Eds.), Proc. 15th IEEE Symp. on Computer Arithmetic, Vail, Colorado, IEEE Computer Society Press, Los Alamitos, CA, 2001, pp. 147–154.
[5] J. Esmonde, J.R. Murty, M.R. Murty, Problems in Algebraic Number Theory, in: Graduate Texts in Mathematics, Vol. 190, Springer, New York, 1999.
[6] A. Granville, T.J. Tucker, It's as easy as *abc*, Notice Amer. Math. Soc. 49 (10) (2002) 1124–1231.
[7] M. Hindry, J.H. Silverman, Diophantine Geometry: An Introduction, in: Graduate Texts in Mathematics, Vol. 201, Springer, New York, 2000.
[8] C.S. Iordache, D.W. Matula, Infinitely precise rounding for division, square root, and square root reciprocal, in: I. Koren, P. Kornerup (Eds.), Proc. 14th IEEE Symp. on Computer Arithmetic, Adelaide, Australia, IEEE Computer Society Press, Los Alamitos, CA, 1999, pp. 233–240.
[9] IBM, Technical Reference: Base Operating System and Extensions, 4th Edition, Vol. 2, International Business Machines Corporation, 2002, available at http://www16.boulder.ibm.com/cgi-bin/ds_rslt1.
[10] S. Lang, Algebra, 3rd Edition, Addison-Wesley Pub Co, Reading, MA, 1992.
[11] T. Lang, J.-M. Muller, Bounds on runs of zeros and ones for algebraic functions, in: N. Burgess, L. Ciminiera (Eds.), Proc. 15th IEEE Symp. Computer Arithmetic, Vail, Colorado, IEEE Computer Society Press, Los Alamitos, CA, 2001, pp. 13–20.
[12] R.-C. Li, P. Markstein, J. Okada, J. Thomas, The `libm` library and floating-point arithmetic in HP-UX for Itanium II, available at http://h21007.www2.hp.com/dspp/files/unprotected/Itanium/FP_White_Paper_v2.pdf (June 2002).
[13] J. Liouville, Sur des classes très étendues de quantités dont la valeur n'est ni algébrique, ni même réductible à des irrationnelles algébriques, C.R. Acad. Sci. Paris Sér. A 18 (1844) 883–885.
[14] J. Liouville, Nouvelle démonstration d'un théorème sur les irrationnelles algébriques inséré dans le compte rendu de la dernière séance, C.R. Acad. Sci. Paris Sér. A 18 (1844) 910–911.

[15] J. Liouville, Sur des classes très étendues de quantités dont la valeur n'est ni algébrique, ni même réductible à des irrationnelles algébriques, J. Math. Pures Appl. 16 (1851) 133–142.
[16] B. Mazur, Questions about powers of numbers, Notice Amer. Math. Soc. 47 (2) (2000) 195–202.
[17] K.F. Roth, Rational approximations to algebraic numbers, Mathematika 2 (1955) 1–20.
[18] A. Wiles, Modular elliptic curves and Fermat's last theorem, Ann. Math. 141 (1995) 443–551.